

Ferdinand Lehr's Weblog

[Startseite](#)

[Kunstgalerie](#)

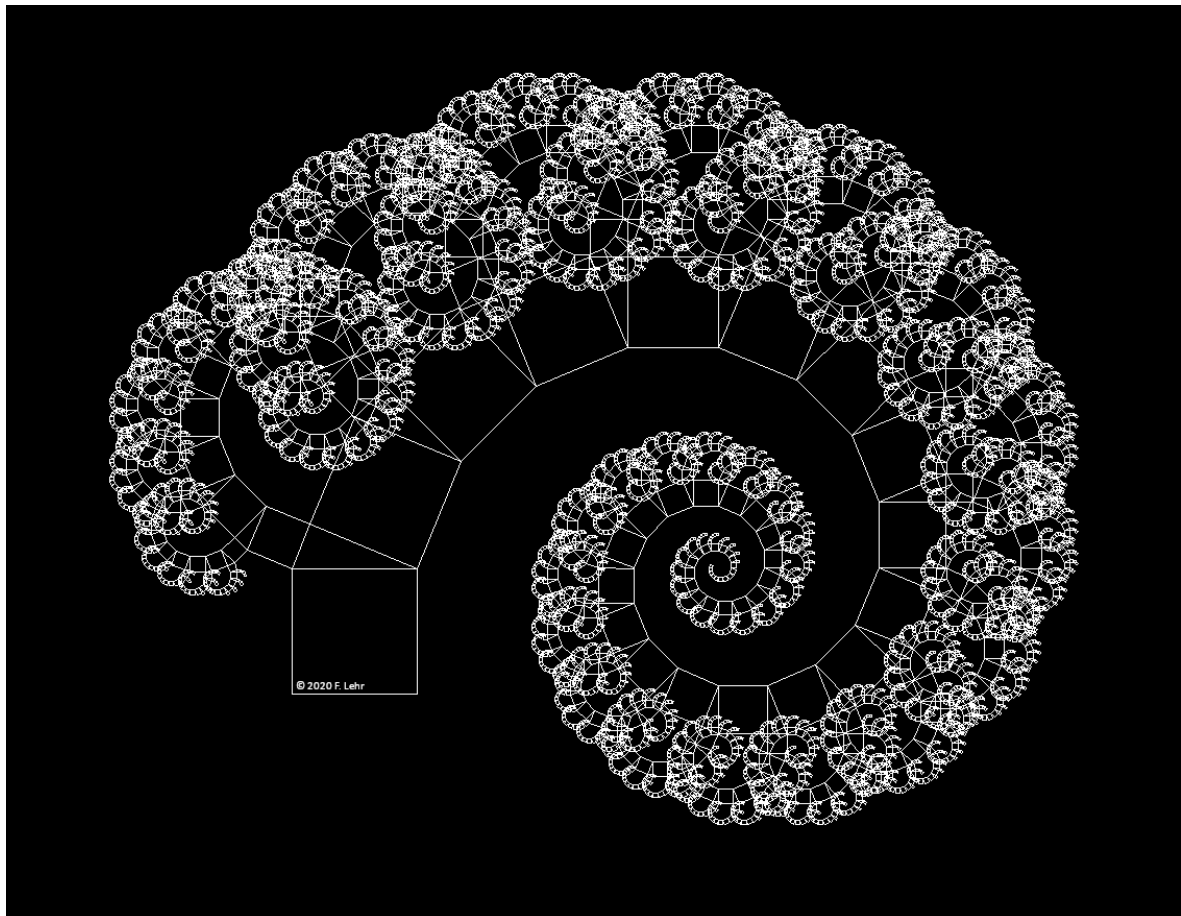
[Fotografie](#)

[Kontakt](#)

[Datenschutzerklärung](#)

[Impressum](#)

Fer



Pythagoras-Baum

3. April 2021 von Ferdinand Lehr

Der Pythagoras-Baum wird gebildet durch rekursive Fortsetzung der Gleichung $a^2+b^2=c^2$ und deren grafische Darstellung mit einem Computerprogramm. Ausgehend von c^2 als Baumstamm verzweigen sich die Äste jeweils nach a^2 und b^2 . Jeder dieser beiden Zweige wird dann im nächsten Durchlauf zu c^2 .

Quellcode (FreeBASIC):

```
'-----'
'          PYTHAGORAS-BAUM          '
'      (c) 2020 F. Lehr              '
'  written for FreeBASIC-Compiler 1.05.0  '
'-----'
```

```
Const PI As Double = 3.1415926535897932
```

```
type tTeil
```

```
    dim as single  x1=0,y1=0,_
                    x2=0,y2=1,_
                    x3=1,y3=1,_
                    x4=1,y4=0,_
                    rx=0,ry=1,_
                    mx=0.5,my=1,_
                    a=30,_
                    self_angle=0,_
                    xo=0,yo=0,_
                    scale=1
```

```
end type
```

```
dim shared zoom as single = 2
```

```
dim shared xoffs as single = 1024/2 -100
```

```
dim shared yoffs as single = 768/2 +20
```

```
declare function GetX1(t as tTeil) as single
declare function GetY1(t as tTeil) as single
declare function GetX2(t as tTeil) as single
declare function GetY2(t as tTeil) as single
declare function GetX3(t as tTeil) as single
declare function GetY3(t as tTeil) as single
declare function GetX4(t as tTeil) as single
declare function GetY4(t as tTeil) as single
declare function GetRPX(t as tTeil) as single
declare function GetRPY(t as tTeil) as single
```

```
Function RotatePoint(ByRef x As single, ByRef y As single, ByVal w ,
```

```
    w = w * PI / 180.0
```

```
    Dim As single x2,y2
```

```
    x2 = x*cos(w) - y*sin(w)
```

```
    y2 = x*sin(w) + y*cos(w)
```

```
    x=x2
```

```
    y=y2
```

```

        return 0
End Function

function GetDistance(x1 as single, y1 as single, _
                    x2 as single, y2 as single) as single
    dim as single r
    r = sqr((x2-x1)^2 + (y2-y1)^2)
    return r
end function

sub ZeichneTeil(t as tTeil)

    Line (GetX1(t)*zoom+xoffs,-GetY1(t)*zoom+yoffs)-_
        (GetX2(t)*zoom+xoffs,-GetY2(t)*zoom+yoffs),15
    Line (GetX2(t)*zoom+xoffs,-GetY2(t)*zoom+yoffs)-_
        (GetX3(t)*zoom+xoffs,-GetY3(t)*zoom+yoffs),15
    Line (GetX3(t)*zoom+xoffs,-GetY3(t)*zoom+yoffs)-_
        (GetX4(t)*zoom+xoffs,-GetY4(t)*zoom+yoffs),15
    Line (GetX4(t)*zoom+xoffs,-GetY4(t)*zoom+yoffs)-_
        (GetX1(t)*zoom+xoffs,-GetY1(t)*zoom+yoffs),15

end sub

sub SetTeilRotation(ByRef t as tTeil, w as single)
    t.self_angle = w
end sub

sub SetRPWinkel(ByRef t as tTeil, w as single)
    t.a = w
end sub

sub SetTeilVerschiebung(ByRef t as tTeil, x as single, y as single)
    t.xo=x
    t.yo=y
end sub

sub SetTeilSkalierung(ByRef t as tTeil, s as single)
    t.scale=s
end sub

function GetX1(t as tTeil) as single

```

```

    dim as single x=t.x1,y=t.y1
    RotatePoint(x,y,t.self_angle)
    x=x*t.scale+t.xo
    y=y*t.scale+t.yo
    return x
end function

function GetY1(t as tTeil) as single
    dim as single x=t.x1,y=t.y1
    RotatePoint(x,y,t.self_angle)
    x=x*t.scale+t.xo
    y=y*t.scale+t.yo
    return y
end function

function GetX2(t as tTeil) as single
    dim as single x=t.x2,y=t.y2
    RotatePoint(x,y,t.self_angle)
    x=x*t.scale+t.xo
    y=y*t.scale+t.yo
    return x
end function

function GetY2(t as tTeil) as single
    dim as single x=t.x2,y=t.y2
    RotatePoint(x,y,t.self_angle)
    x=x*t.scale+t.xo
    y=y*t.scale+t.yo
    return y
end function

function GetX3(t as tTeil) as single
    dim as single x=t.x3,y=t.y3
    RotatePoint(x,y,t.self_angle)
    x=x*t.scale+t.xo
    y=y*t.scale+t.yo
    return x
end function

function GetY3(t as tTeil) as single
    dim as single x=t.x3,y=t.y3
    RotatePoint(x,y,t.self_angle)
    x=x*t.scale+t.xo

```

```

        y=y*t.scale+t.yo
        return y
end function

function GetX4(t as tTeil) as single
    dim as single x=t.x4,y=t.y4
    RotatePoint(x,y,t.self_angle)
    x=x*t.scale+t.xo
    y=y*t.scale+t.yo
    return x
end function

function GetY4(t as tTeil) as single
    dim as single x=t.x4,y=t.y4
    RotatePoint(x,y,t.self_angle)
    x=x*t.scale+t.xo
    y=y*t.scale+t.yo
    return y
end function

function GetRPX(t as tTeil) as single
    dim as single x=0.5,y=0
    RotatePoint(x,y,180-t.a)
    x=x+0.5
    y=y+1
    RotatePoint(x,y,t.self_angle)
    x=x*t.scale+t.xo
    y=y*t.scale+t.yo
    return x
end function

function GetRPY(t as tTeil) as single
    dim as single x=0.5,y=0
    RotatePoint(x,y,180-t.a)
    x=x+0.5
    y=y+1
    RotatePoint(x,y,t.self_angle)
    x=x*t.scale+t.xo
    y=y*t.scale+t.yo
    return y
end function

function GetAngleA(t as tTeil) as single

```

```

        return (180-t.a)/2
end function

function GetAngleB(t as tTeil) as single
    return (180-90)-GetAngleA(t)
end function

function GetDistanceA(t as tTeil) as single
    return GetDistance(GetX2(t),GetY2(t),GetRPX(t),GetRPY(t))
end function

function GetDistanceB(t as tTeil) as single
    return GetDistance(GetX3(t),GetY3(t),GetRPX(t),GetRPY(t))
end function

sub ZeichneBaum(t as tTeil, _
                wrot as single, _
                winkel as single, _
                skalierung as single, _
                x as single, y as single, _
                depth as integer)
    if depth<1 or skalierung<1 then
        exit sub
    end if

    SetRPWinkel(t,winkel)
    SetTeilRotation(t,wrot)
    SetTeilSkalierung(t,skalierung)
    SetTeilVerschiebung(t,x,y)
    ZeichneTeil(t)

    dim as tTeil t2, t3
    ZeichneBaum(t2, wrot+GetAngleA(t), winkel, GetDistanceA(t), _
                GetX2(t), GetY2(t), depth-1)
    ZeichneBaum(t2, wrot-GetAngleB(t), winkel, GetDistanceB(t), _
                GetRPX(t), GetRPY(t), depth-1)

end sub

'Hauptprogramm

screen 20

```

```
dim t as tTeil
```

```
ZeichneBaum(t, 0, 45, 60, -100, -100, 50)
```

```
sleep
```

```
'----- PROGRAMMENDE -----'
```

NEUESTE BEITRÄGE



Sternenschanze Lienzingen/Ötisheim



Neue Fotogalerie



"Hanussen II" (Willi Gerstel)

NEU IN DER GALERIE

